

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

**REMARKS**

**INTRODUCTION**

Claims 1-4, 6-16, and 18-33 were previously and are currently pending and under consideration.

Claims 1-4, 6-16, and 18-33 are rejected.

Claims {to be determined, at least all indep claims} are amended herein.

No new matter is being presented, and approval and entry are respectfully requested.

**ENTRY OF AMENDMENT UNDER 37 CFR §1.116**

Applicant requests entry of this Rule 116 Response because:

- (a) it is believed that the amendment of the claims puts this application into condition for allowance;
- (b) the amendments were not earlier presented because the Applicant believed in good faith that the cited prior art did not disclose the present invention as previously claimed;
- (c) the amendments of the claims should not entail any further search by the Examiner since no new features are being added or no new issues are being raised; and
- (d) the amendments do not significantly alter the scope of the claims and place the application at least into a better form for purposes of appeal. No new features or new issues are being raised.

The Manual of Patent Examining Procedures sets forth in Section 714.12 that "any amendment that would place the case either in condition for allowance or in better form for appeal may be entered." Moreover, Section 714.13 sets forth that "the Proposed Amendment should be given sufficient consideration to determine whether the claims are in condition for allowance and/or whether the issues on appeal are simplified." The Manual of Patent Examining Procedures further articulates that the reason for any non-entry should be explained expressly in

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

the Advisory Action.

**REJECTIONS UNDER 35 USC § 103**

In the Office Action, claims 1-4, 6-16, and 18-33 were rejected under 35 U.S.C. § 103 as being unpatentable over Lumelsky in view of Ding. This rejection is traversed and reconsideration is requested.

Lumelsky Does Not Signal When Prediction Fails To Meet Availability Threshold

Claim 1 recites "providing a signal when said prediction of the future level of availability of the monitored resource fails to meet an availability threshold". The rejection cites column 16, lines 9-37 of Lumelsky as teaching this feature. However, as discussed below, the cited portion of Lumelsky is nothing more than the ordinary real-time or immediate threshold-exceeding feature found in many operating systems, where an alert is raised when a current resource level reaches a critical threshold. No resource prediction is involved.

The cited portion begins by stating that "FIG. 8(b) is a flow chart depicting ... the real-time resource monitor process ... this functionality is standard in most computer operating systems". In other words, the cited portion relates to real-time monitoring. Lumelsky then mentions examples of resources monitored, such as buffer management, VM page hits, etc. Modern operating systems typically provide real-time data on these types of resources. Lumelsky then states that "requirements departure [e.g. threshold violation] is estimated, e.g., the number of I/O buffers needed to stream (e.g., 1 MB) and bandwidth". This is nothing more than computing thresholds to be used for monitoring. The next sentence in Lumelsky explains that there are techniques such as *video smoothing* that "allow estimating reliably these values and determining under/over flow conditions." In other words, performance thresholds can be reliably estimated because the type of computing (*video smoothing*) is known. To this point, the cited portion has only discussed real-time monitoring of resources and reliably estimating thresholds for monitoring.

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

The cited portion next discusses smoothing monitored data. Smoothing has nothing to do with prediction of future resource availability. Rather, smoothing is a well known statistical technique where sampled data point magnitudes are readjusted, for example to lessen the impact of anomalies or to locally average values. Smoothing sampled resource usage data does not predict whether that resource data will increase or decrease in the future. It only readjusts measurements so that the data points form a smooth graph rather than a set of data points that vary drastically from one measurement data point to the next. It is well known that operating systems provide this type of smoothing of past sampled performance data. For example, in a Windows XP CPU monitor window, the CPU usage curve may appear smooth, although in reality the CPU usage may rise and fall very rapidly over very small time increments. In sum, smoothing monitoring resource usage data does not teach predicting availability of the resource in the future.

The last part of the cited portion of Lumelsky (column 16, lines 25-37) relates to "[t]rend assessment". This is described as involving "A critical threshold ... associated with each particular resource may be used to determine whether any departure is statistically significant", in which case an exception is generated. If an exception occurs, "then the resource allocation is compensated, as indicates at step (885)". This last portion of Lumelsky relates only to generating an exception when a resource significantly departs in real-time from its threshold; departure is a current not future occurrence, and no prediction is mentioned or suggested.

The meaning of the cited portion is also apparent in Figure 8(b), to which the cited portion directly corresponds. Step 875 is to "Estimate Requirements Departure". Rather than predicting a future usage or availability, the cited portion of Lumelsky only relates to estimating the value at which an exception occurs in real-time. As stated at the beginning, the cited portion relates to a "real-time resource monitor process" which makes sense because Lumelsky is concerned with ensuring that a media server has sufficient resources for video streaming, which is inherently real-time critical. Note also that Lumelsky describes detecting exceptions with certainty, however, if Lumelsky were making future threshold projections, its detecting would most likely be described in less certain terms; projections or predictions are not certain. Lumelsky's real-time resource monitoring and current/real-time departure determination is

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

significantly different than comparing a threshold to a future predicted availability of a resource, as in claim 1.

The rejection also cites column 2, line 65, to column 3, line 1 of Lumelsky. This portion of Lumelsky states that "It is essential that quality of service is configurable, predictable and maintainable system-wide." According to Newton's Telecom Dictionary (20th ed.), Quality of Service (QoS) "is a measure of the telecommunications – voice, data and/or video – service quality provided to a subscriber. ... multimedia streams, such as those used in IP telephone or videoconferencing, may be extremely bandwidth and delay sensitive, imposing unique quality of service (QoS) demands ... In order to deploy real-time applications over IP networks with an acceptable level of quality, certain bandwidth, latency, and jitter requirements must be guaranteed". Therefore, it is clear that the cited port of columns 2-3 is only a restatement of the general purpose of Lumelsky. Because QoS is predictable does not imply that there is a trigger of allocation of a physical resource to be manually added when a predicted future availability of a physical resource reaches a threshold. Continued Quality of Service in Lumelsky is predictable because, for example, a new resource request is denied if it would affect the quality of current services. Furthermore, QoS is not the same as a physical resource, rather it is a measure of service, not hardware.

Column 14, lines 1-21 are also cited. However, this portion of Lumelsky describes that when a new service is introduced, a resource envelope, from the creator of the service, is distributed. "If a provisioning request is dispatched to a meta-resource, the service unit management module (SUMM) determines whether the provisioning request can be scheduled by the meta-resource given the available resources. If a reservation for a number of service units is requested to the meta-resource, the SUMM determines whether the reservation can be scheduled given the projected resources." In the case of a provisioning request, Lumelsky is clearly making a real-time current-availability determination, which clearly is not the claimed feature. In the case of the reservation request, Lumelsky is only deciding whether to grant a reservation request. In this case, there is a reservation request and a determination if there will be sufficient resources therefor. In some ways, this is the opposite order of the recited claim feature. The recited claim feature first determines a future/projected shortcoming of a resource

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

and in response automatically reserves additional physical resources to be manually added. In contrast, Lumelsky first makes an allocation/reservation request *and then* determines whether resources will be available to meet the reservation request. If there are not enough projected resources then the reservation is not granted. In other words, Lumelsky is only deciding whether to schedule a reservation of a given resource. Again, Lumelsky teaches run-time compensation ("monitoring of resource reservation exceptions ... [which are forwarded] to the run-time compensation unit", which "triggers a compensation of the resource envelope for a service unit during run-time", column 14, lines 36).

Although not cited for the feature discussed above, Ding similarly describes raising alerts "whenever usage of a particular software process exceeds a particular threshold" (column 7, lines 42-48).

Claims 9, 15, 23, and 29 recite similar features.

Withdrawal of the rejection is respectfully requested.

**Ding Does Not Automatically Allocate Resources**

Ding is cited for teaching automatically allocating an additional hardware resource to be manually physically added to the computer. However, as explained below, Ding does not automatically allocate an additional hardware resource. Rather, in Ding a user analyzes the output of Ding and then the user enters input to alter the system configuration. Therefore, Ding does not disclose automatic resource allocation. Furthermore, as discussed further below, altering a "configuration" does not allocate actual physical resources.

The rejection cites column 21, line 53, to column 22, line 4 as teaching this feature. This portion states (emphasis added):

In one embodiment, the enterprise is modeled and/or its configuration is altered in response to the determination(s) of utilization described herein. Modeling according to one embodiment is discussed in detail with reference to FIGS. 6 and 7. In various embodiments, this modeling may further comprise one or more of the following: displaying the determination(s) to a user, predicting future performance, graphing a performance prediction, generating reports, asking a user for further data, and permitting a

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

user to modify a model of the enterprise. In one embodiment, Analyze 406 and/or Predict 408, as discussed in detail with reference to FIGS. 6 and 7, implement the modeling, analysis, and/or prediction in response to the determination(s) of utilization. In one embodiment, a configuration of the enterprise is altered in response to the determination(s) of utilization. Altering a configuration of the enterprise may comprise, for example, reconfiguring a network topology or installing additional resources, such as CPUs, software, memory resources, or network routers or hubs.

The altering discussed in this portion is performed by a user. Figure 7 shows that "User Input: Configuration Corrections[,] Configuration Changes[, and] Growth Scenarios" are inputted by the user and passed to the Predict 408 module. The rejection compares altering a configuration to allocation of resources. However, the configuration is changed by a user. At column 3, lines 47-57, Ding summarizes "displaying the determinations to a user, predicting future performance, graphing a performance prediction, generating reports, asking a user for further data, permitting a user to modify a model of the enterprise, and altering a configuration of the enterprise in response to the determinations". Furthermore, in discussing Figure 6 (referred to in the portion cited by the Examiner), Ding states that "With Visualizer 410, performance statistics and workloads can be graphed, compared, drilled down, and visually analyzed to pinpoint hot spots or trends to assist in resource management, system tuning, and configuration changes" (column 10, line 64, to column 11, line 1, emphasis added). Also, "After the baseline model has been constructed, the user can- modify the baseline model by specifying configuration corrections, configuration changes, and/or growth scenarios" (column 11, lines 33-35). At column 6, lines 24-27, Ding states that "The enterprise management system 180 permits users to monitor, analyze, and manage resource usage on heterogeneous computer systems 150 across the enterprise 100". In other words, actual management of resources is preformed by users. In explaining the overall console, Ding notes that "The Predict component 408 takes the model from the Analyze component 406 and allows a user to alter the model by specifying hypothetical changes to the enterprise 100" (column 6, lines 7-9).

Withdrawal of the rejection is respectfully requested.

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

Claim 1, for example, also recites "automatically allocating additional resources to be manually physically added to the computer". As discussed in the background of the present specification, before the present invention, resources were manually added long after a need for the same arose, and sometimes the added resources were retained when no longer needed (page 2, lines 4-5). Users would turn off their machine, insert more resources, and restart the machine. Some users do not understand when a new resource is necessary, and so an upgrade would be made only when it became an issue (page 2, lines 14-16). The specification mentions that an aspect of the present invention *reduces* downtime (page 3, lines 18-20) by allocating or reserving a resource when its need is predicted. Memory, for example, is described as being added to or subtracted from a computer system. That is, the present specification discloses a resource being allocated when needed, and the resource being physically added afterwards as a separate step from the allocation. The Merriam Webster Online Dictionary indicates that to "allocate" is to "to set apart or earmark : DESIGNATE <allocate a section of the building for special research purposes>". In Lumelsky, a resource may be allocated, but there is no physical resource addition. Lumelsky has a fixed pool of resources, and an allocation draws on or returns resources to the fixed pool. The allocation itself secures the resource for a service, which is different than an allocation followed by a resource addition. For further understanding, consider the memory example in the specification. If a prediction indicates that memory will run low, the system is requested to allocate additional memory. Later, when the memory has been added as a separate step, the proper charge is made to the user (page 6, line 25, to page 7, line 1). In sum, Lumelsky's allocation itself makes a resource available, whereas claim 1 earmarks reserves the needed resource (which does not make the resource available). The resource becomes available when it is later added to the computer as a separate step.

Withdrawal of the rejection is further respectfully requested.

Dina Does Not Automatically Reserve/Order an Additional Physical Hardware Resource

Amended claim 1 recites "without user intervention, responding to the signal by automatically reserving or ordering an additional physical hardware resource that is not in the computer when the signal is provided and which is to be later manually physically added to the computer after the reserving or placing of an order". The previously pending claims recited

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

"allocating" physical resources.

The rejection cites only column 21, line 53 to column 22 line 4 of Ding, which the rejection characterizes as "installing". Applicant respectfully notes that according to MPEP § 2131, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. ... The identical invention must be shown in as complete detail as is contained in the ... claim" (emphasis added).

The cited portion of Ding simply does not describe any detail of actually automatically reserving or ordering a non-present physical resource to be added later. The cited portion of Ding relates to altering only information, in the form of a model or a "configuration". Ding suggests that "configuration" is only "constant data [that] does not change over the measurement interval or lifetime of the event ... configuration information [etc.] are generally constant values". Also, Ding states that "[a]fter the baseline model has been constructed, the user can modify the baseline model by specifying configuration corrections, configuration changes, and/or growth scenarios. ... Predict 408 accurately determines the impact of these workload and configuration changes on performance and response time. As one of the results of "what if?" computation, the changes to the baseline are displayed" (column 11, lines 33-43). When a user changes a "configuration" they are not reserving or ordering actual physical resources to be added later, rather the user is changing a *model* of the system for the purpose of testing different *hypothetical* ("what if?") configuration scenarios. For example, a user may test a hypothetical installation of memory. This is also seen at column 11, line 63, to column 12 line 4, which states:

in modeling the configuration and workload changes across multiple systems, Predict 408 automatically calculates interaction and interference between systems. Predict 408 also preferably provides scenario planning, or modeling incremental growth over time, in order to determine the life expectancy of computing resources and the point at which resources should be upgraded to ensure that performance remains at an acceptable level. In the various ways set forth above, *Predict 408 thus permits a user to plan for the future by "test driving" both actual and alternative or hypothetical configurations of the enterprise.* 100.

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

In other words, in Ding a user tests different hypothetical configuration models and a user can predict when a future upgrade may be needed. Ding is a planning tool, not a tool for reserving or ordering additional physical resources to be added to a computer. Note that there is no detail or disclosure about automatically reserving or allocating resources not present at the time of determining the future need therefor. Ding is only a planning tool and does not connect predictions with actual reservation of physical resources.

**Improper Combination: Non-Analogous and Incompatible References**

Applicant respectfully notes that Lumelsky is a system for reallocating the same resources in a media server to assure quality of service for different media streams. Ding is a system or enterprise wide analysis tool that collects data from different computers and network devices to monitor networks and computing devices. Lumelsky concerns only the reshifting of the same resources within one computer to different media streams. Ding only monitors and analyzes an enterprise computing environment and has no relation to media streaming. One skilled in the art would not have considered combining these references which have significantly different purposes and areas of operation. The rejection cites column 2, lines 41-44 as providing a motive for the combination. However, the very next sentence says that Ding is for use "in a distributed computing environment, i.e., an enterprise".

The combination is also traversed because it is not sufficiently specific. In *Ex parte Humphreys* (24 USPQ 2d 1255), the Board held that an Examiner must provide specific reasons to support an obviousness rejection. The Board stated that "The examiner's rejection is not specific as to how one of ordinary skill in the art would have found it obvious to practice any specific method within the scope of these claims as of the filing date of this application ... the examiner has not explained with any specificity ... how [the prior art would have suggested the combination]". According to MPEP § 2144, the "examiner must present convincing line of reasoning supporting rejection [and] rely on logic and sound scientific reasoning". The rejection does not include any reasoning or logic explaining why one would have made the specific combination. The motive of "more accurate and efficient monitoring and prediction of computer system performance" is too general and involves no reasoning or explanation beyond restating a benefit of Ding by itself. The specific purpose of Ding is to "accurately and efficiently

**RESPONSE UNDER 37 CFR 1.116  
EXPEDITED PROCEDURE  
EXAMINING GROUP 3677  
Serial No. 09/734,273**

reflect system resource usage at a lower sampling frequency" (column 2, lines 35-38), which has no relation to the modification for which Ding is cited, namely, automatically allocating an additional hardware resource to be manually physically added to the computer.

#### **DEPENDENT CLAIMS**

The dependent claims are deemed patentable due at least to their dependence from allowable independent claims. These claims are also patentable due to their recitation of independently distinguishing features. For example, claim 7 recites "analyzing available applications as a function of at least one computer resource". This feature is not taught or suggested by the prior art. Withdrawal of the rejection of the dependent claims is respectfully requested.

#### **CONCLUSION**

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Respectfully submitted,

Date: 19 JAN 2005

By: James T. Strom  
James T. Strom  
Registration No. 48,702

**CERTIFICATE OF FACSIMILE TRANSMISSION**

I hereby certify that this correspondence is being transmitted via facsimile to: Commissioner for Patents,

P.O. Box 1450, Alexandria, VA 22313-1450

on January 19, 2005

STAAS & HALSEY

By: Jacqueline Anderson

Date: January 19, 2005